LING 331
Text Processing for Linguists

Week 4

Basic Python 2
Notes from Assignment 3

Biggest thing:

*please* make sure your assignment runs all the way through on Quest!
Notes from Assignment 3

- Core ways to read files:
  ```python
  for line in open(f):
    vs
  text = open(f).read()
  ```

- You can do operations in defining `for` loops, e.g.:
  ```python
  for word in s.split():
  ```

- `continue` statement in loops
Meaningful variable names!

- Be aware of clashes - can still work but be confusing!

```python
def mean(vals):
    mean = 0
    for val in vals:
        mean += val
    return mean / len(vals)
```
Scope determines where objects are defined

```python
# `print` is built-in
print('hi!')

# non-indented is global
my_var = 'hellooooo'

def my_func():
    # only available
    # inside the function
    local_var = 'hey?'

# gets NameError
print(local_var)
```

[Diagram of namespaces and scope](https://www.geeksforgeeks.org/namespaces-and-scope-in-python/)
Sets are unordered collections of unique elements

Analogous to sets in math, lists of unique items

<table>
<thead>
<tr>
<th>Set Operation</th>
<th>Venn Diagram</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Union</td>
<td><img src="https://www.datacamp.com/community/tutorials/sets-in-python" alt="Union Diagram" /></td>
<td>$A \cup B$, is the set of all values that are a member of $A$, or $B$, or both.</td>
</tr>
<tr>
<td>Intersection</td>
<td><img src="https://www.datacamp.com/community/tutorials/sets-in-python" alt="Intersection Diagram" /></td>
<td>$A \cap B$, is the set of all values that are members of both $A$ and $B$.</td>
</tr>
<tr>
<td>Difference</td>
<td><img src="https://www.datacamp.com/community/tutorials/sets-in-python" alt="Difference Diagram" /></td>
<td>$A \setminus B$, is the set of all values of $A$ that are not members of $B$.</td>
</tr>
<tr>
<td>Symmetric Difference</td>
<td><img src="https://www.datacamp.com/community/tutorials/sets-in-python" alt="Symmetric Difference Diagram" /></td>
<td>$A \triangle B$, is the set of all values which are in one of the sets, but not both.</td>
</tr>
</tbody>
</table>

https://www.datacamp.com/community/tutorials/sets-in-python
Set Methods

s = set()  # create an empty set
s.add(val)  # add a value to the set
s.remove(val)  # remove a value from the set
s1 & s2  # set intersection
s1 - s2  # set difference
s1.issubset(s2)  # set operations
s1.issuperset(s2)
s1.union(s2)
s1.intersection(s2)

len(s)  # number of items in the set
Dictionaries define key-value mappings

Versatile mappings between (almost) whatever and whatever else

Dict keys must:

- Be immutable
- Appear only once

https://developers.google.com/edu/python/dict-files
Dictionary Methods

d = {}  # create an empty dictionary and assign it to d

d[key] = value  # assign a value to a given dictionary key

d.keys()  # the list of keys of the dictionary

d.values()  # the list of values in the dictionary

if key in d:  # test whether a particular key is in the dictionary

for key in d:  # iterate over the keys of the dictionary

len(d)  # number of keys in the dictionary
Random Built-in Module

We’ll use it a lot in this assignment!

`random.random()` with nested conditionals:

```python
randval = random.random()  # float between 0 and 1
if randval < 0.35:
    # 35% chance of entering here
elif randval < 0.6:
    # 25% chance of entering here
else:
    # 40% chance
```

**Diagram:**
- **Orange** region: `if randval < 0.35`
- **Blue** region: `elif randval < 0.6`
- **Green** region: `else`

**Example Result:**
```
randval == 0.53462123213
```